

Grid-based electronic structure calculations: the tensor decomposition approach

M.V. Rakhuba^a, I.V. Oseledets^{a,b}

^aSkolkovo Institute of Science and Technology, Novaya St. 100, 143025 Skolkovo, Moscow Region, Russia.

^bInstitute of Numerical Mathematics, Russian Academy of Sciences. Gubkina St. 8, 119333 Moscow, Russia.

Abstract

We present a fully grid-based approach for solving Hartree-Fock and all-electron Kohn-Sham equations based on low-rank approximation of three-dimensional electron orbitals. Due to the low-rank structure the total complexity of the algorithm depends linearly with respect to the one-dimensional grid size. Linear complexity allows for the usage of fine grids, e.g. 8192^3 and, thus, cheap extrapolation procedure.

We test the proposed approach on closed-shell atoms up to the argon, several molecules and clusters of hydrogen atoms. All tests show systematical convergence with the required accuracy.

1. Introduction

Electronic structure calculations arise in a variety of applications such as condensed matter physics or material and drug design. In this paper we focus on the development of efficient and accurate numerical techniques for solving Hartree-Fock (HF) [1] and Kohn-Sham (KS) equations [2]. From computational point of view HF and KS equations are nonlinear eigenvalue problems with three-dimensional integro-differential operators. A standard way to solve these equations is to approximate the solution in a subspace of globally supported basis functions, e.g. gaussian functions. This is a classic topic and a lot of software packages are available. The choice of these basis functions is determined by the complexity of the iterative process and the desired accuracy. It also introduces the *basis set error*, which might be difficult to control. Methods based on a sequence of embedded subspaces allow for rigorous error control. Among these approaches we should mention multiresolution approach based on wavelet function representation [3], finite element methods on unstructured grids [4], finite difference method [5] and the projector-augmented wave method [6]. Standard finite element or finite difference method on uniform grids for a 3D-dimensional problem has $O(n^3 \log n)$ complexity (using Fast Fourier Transform for the evaluation of the integral operators). On non-uniform meshes we can not use FFT directly, thus the complexity can reach $O(n^6)$, where n is the one-dimensional grid size.

One of the promising ways to reduce the complexity, proposed and studied in details in the papers by Khoromskij and Khoromskaia [7, 8, 9, 10, 11, 12, 13, 14, 15] is to use *tensor decomposition* of the vectors of coefficients. It was shown that in many cases, the tensor of coefficients can be well-approximated by the *Tucker decomposition*, and the storage complexity

Email addresses: rakhuba.m@gmail.com (M.V. Rakhuba), i.oseledets@skoltech.ru (I.V. Oseledets)

Preprint submitted to Elsevier

September 1, 2015

goes down to $O(n \log n)$, where n is one-dimensional grid-size. This allows for huge uniform grids to be used. However, to compute the solution, they still employ an intermediate basis set (which can be arbitrary), and successively compute the Fock matrix and update the coefficients with respect to this basis. The main difficulty in solving the KS/HF equation directly in the tensor format is that standard iterative methods should be modified in order to keep the solution in the tensor format, and it requires the development of a new iterative method.

Our main contribution in this paper is an efficient $O(n \log n)$ black-box solver which is based on low-rank tensor decompositions, and it does not require any additional basis sets. Thus, it can be used to control the basis set error in HF/KS computations. We have developed an iterative process that solves the original problem while all intermediate three-dimensional arrays of coefficients are stored in the Tucker format. Our method has several important components: we use the block integral iteration, fast low-rank convolution and derivative-free update of the Fock matrix. Finally, the extrapolation over several grid sizes is done, giving up to $O(h^4)$ accuracy, where h is the grid step.

We organize our paper as follows. In section 2 we formulate the closed-shell Hartree-Fock and Kohn-Sham equations. In section 3 we discuss tensor decomposition approach and the Tucker format. Section 4 contains formulation of the Block-Green iteration and the derivative-free Fock matrix update formula. Sections 5 and 6 are concerned with obtaining the discrete formulation and all operations involved in the tensor formats. In section 7 we discuss the complexity of the proposed algorithm. Finally, in section 8 we present numerical experiments on closed-shell atoms up to argon, several molecules and cluster of hydrogen atoms, which illustrate sublinear rank dependence.

2. Hartree-Fock and Kohn-Sham equations

Consider closed-shell electron system with N_e electrons and $N = N_e/2$ orbitals, N_{nuc} nuclei with charges Z_α located at \mathbf{R}_α , $\alpha = 1, \dots, N_{nuc}$. Then the HF/KS equations can be written as [16]

$$H(\Phi) \phi_i \equiv \left(-\frac{1}{2} \Delta + V(\Phi) \right) \phi_i = \lambda_i \phi_i, \quad i = \overline{1, N} \quad (1)$$

where $\phi_i : \mathbb{R}^3 \rightarrow \mathbb{R}$ denotes unknown orbitals that additionally satisfy the orthogonality constraints

$$\int_{\mathbb{R}^3} \phi_i(\mathbf{r}) \phi_j(\mathbf{r}) d\mathbf{r} = \delta_{ij}.$$

and λ_i denotes orbital energies. In the Kohn-Sham model

$$V(\Phi) \equiv \tilde{V}(\rho) = V_{ext} + V_{coul}(\rho) + V_{xc}(\rho), \quad (2)$$

where

$$\rho(\mathbf{r}) = 2 \sum_{i=1}^N |\phi_i(\mathbf{r})|^2,$$

is the electron density. The external potential V_{ext} describes interaction between electrons and nuclei:

$$V_{ext}(\mathbf{r}) = - \sum_{\alpha=1}^{N_{nuc}} \frac{Z_\alpha}{|\mathbf{r} - \mathbf{R}_\alpha|}. \quad (3)$$

The potential $V_{coul}(\mathbf{r})$ is given as

$$V_{coul}(\mathbf{r}) = \int_{\mathbb{R}^3} \frac{\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r}',$$

and V_{xc} depends only on the density and is responsible for exchange and correlation parts of the operator. For the sake of simplicity we consider local density approximation (LDA) with Perdew and Zunger type functional [17]. We note that the concept can be extended to local spin density approximation (LSDA) or more accurate functionals.

In the HF equation the potential $V(\Phi)$ has the form

$$V(\Phi) = V_{ext} + V_{coul}(\rho) - \hat{K}(\Phi),$$

where

$$\hat{K}(\Phi) \phi_i = \int_{\mathbb{R}^3} \frac{\tau(\mathbf{r}, \mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} \phi_i(\mathbf{r}') d\mathbf{r}' \quad \text{with} \quad \tau(\mathbf{r}, \mathbf{r}') = \sum_{i=1}^N \phi_i(\mathbf{r}) \phi_i(\mathbf{r}').$$

As \hat{K} depends not only on density $\rho(\mathbf{r})$, but on all the orbitals explicitly, the HF equation is computationally more expensive in comparison with the DFT analogues. The Fock matrix $\mathbf{F} = \mathbf{F}(\Phi)$ is defined as

$$F_{\alpha\beta} = \int_{\mathbb{R}^3} \phi_\alpha H(\Phi) \phi_\beta d\mathbf{r}, \quad \alpha, \beta = \overline{1, N}.$$

It is diagonal when Φ is the exact solution of (1). Total energy can be calculated as

$$E = 2 \sum_{i=1}^N \lambda_i - \frac{1}{2} \iint_{\mathbb{R}^6} \frac{\rho(\mathbf{r})\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r}d\mathbf{r}' + \iint_{\mathbb{R}^6} \frac{|\tau(\mathbf{r}, \mathbf{r}')|^2}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r}d\mathbf{r}', + E_{nn}$$

for the Hatree-Fock model and

$$E = 2 \sum_{i=1}^N \lambda_i - \frac{1}{2} \iint_{\mathbb{R}^6} \frac{\rho(\mathbf{r})\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r}d\mathbf{r}' + E_{xc}(\rho) + E_{nn},$$

for the Kohn-Sham model, where

$$E_{nn} = \sum_{i=1}^N \sum_{j>i}^N \frac{Z_i Z_j}{|\mathbf{R}_i - \mathbf{R}_j|},$$

describes the repulsion between the nuclei.

3. Tensor decomposition approach

Orbitals $\phi_\alpha, \alpha = 1, \dots, N$ depend on three variables. Discretization on a rectangular $n \times n \times n$ mesh¹ gives a three-dimensional tensor of coefficients. The storage complexity grows cubically in n . To get linear complexity $\mathcal{O}(n)$ we will use Tucker approximation [18, 19, 20] of the all the tensors arising in the computations.

¹the mode sizes can be in general different, but for simplicity we always assume that they are equal to n .

Tensor $A = \{a_{ijk}\}_{i,j,k=1}^n$ is said to be in the Tucker format if it is represented as

$$a_{ijk} = \sum_{\alpha=1}^{r_1} \sum_{\beta=1}^{r_2} \sum_{\gamma=1}^{r_3} g_{\alpha\beta\gamma} u_{i\alpha} v_{j\beta} w_{k\gamma}, \quad (4)$$

where the minimal possible numbers r_1, r_2, r_3 required to represent A in the form (4) are called Tucker ranks, matrices $U = \{u_{i\alpha}\}_{i=1, \alpha=1}^{n, r_1}$, $V = \{v_{j\beta}\}_{j=1, \beta=1}^{n, r_2}$, $W = \{w_{k\gamma}\}_{k=1, \gamma=1}^{n, r_3}$ are called factors, $G = \{g_{ijk}\}_{i,j,k=1}^n$ is called core of the decomposition.

Decomposition (4) contains $r^3 + 3nr$ parameters, compared with n^3 elements of the whole tensor. In practice, instead of the decomposition we use the approximation with some accuracy ϵ .

Thus, each orbital on the grid is approximated in the Tucker format

$$\phi_p(x_i, y_j, z_k) \approx \sum_{\alpha=1}^{r_1(p)} \sum_{\beta=1}^{r_2(p)} \sum_{\gamma=1}^{r_3(p)} g_{\alpha\beta\gamma}^p u_{i\alpha}^p v_{j\beta}^p w_{k\gamma}^p + O(\epsilon), \quad p = \overline{1, N} \quad i, j, k = \overline{1, n}$$

Our main assumption is that the Tucker ranks are small and grow only logarithmically both in n and ϵ^{-1} . This has been verified before in [9]. Once the approximation is computed, we can check the accuracy of the computations by computing relevant physical quantities and comparing it with the results, obtained by other methods.

4. Iterative method

The standard iterative technique to solve HF/KS equations is the self-consistent field iteration (SCF) [16]:

$$H^{(k)} \phi_i^{(k+1)} = \lambda_i^{(k+1)} \phi_i^{(k+1)}, \quad i = \overline{1, N}.$$

This is not easy to implement in the Tucker format, which is a non-linear parametrization of the 3D-tensor, thus the original convex minimization problem is replaced by a non-convex one. Hence, SCF iteration is not much simpler than the original problem. Therefore, we use a more convenient for our purposes Block Green iteration for Lippman-Schwinger form of (1) as it can be easily done within tensor arithmetics. In next subsection we describe the Block Green iteration technique with rotation of orbitals and present formulas to calculate the Fock matrix without direct Laplacian operation.

4.1. The Block Green iteration

Let us rewrite (1) as follows

$$\phi_i = -2(-\Delta - 2\lambda_i)^{-1} V \phi_i.$$

The action of $(-\Delta - 2\lambda_i)^{-1}$ can be written as a convolution with the Yukawa potential

$$(-\Delta - 2\lambda_i)^{-1} V \phi(\mathbf{r}) \equiv \int_{\mathbb{R}^3} \frac{e^{-\sqrt{-2\lambda_i} \|\mathbf{r} - \mathbf{r}'\|}}{4\pi \|\mathbf{r} - \mathbf{r}'\|} V \phi_i(\mathbf{r}') d\mathbf{r}'. \quad (5)$$

We found that direct application of (5) leads to numerical difficulties, which will be discussed later. In this case much more efficient is to solve the screened Poisson equation directly using finite difference method.

For simplicity we start our description of the iteration for the system with one orbital $\phi_1 \equiv \phi$. In this case the k -th step of the Green iteration is

$$\tilde{\phi} = 2 \left(-\Delta - 2\lambda^{(k)} \right)^{-1} V^{(k)} \phi^{(k)}, \quad \phi^{(k+1)} = \tilde{\phi} / \|\tilde{\phi}\|.$$

The energy is calculated as

$$\lambda^{(k+1)} = \left(H^{(k+1)} \phi^{(k+1)}, \phi^{(k+1)} \right). \quad (6)$$

The Hamiltonian $H^{(k)} \equiv H(\phi^{(k)})$ contains the Laplacian Δ . The approximation error in the Tucker format is controlled only in the discrete L_2 norm, thus discrete differentiation will amplify the error. To avoid it (6) can be rewritten as

$$\lambda^{(k+1)} = \lambda^{(k)} + \frac{\left(V^{(k+1)} \tilde{\phi} - V^{(k)} \phi^{(k)}, \tilde{\phi} \right)}{\left(\tilde{\phi}, \tilde{\phi} \right)}. \quad (7)$$

The generalization to the case of molecules with more than one orbital is as follows. We start by modifying each orbital separately

$$\hat{\phi}_i = 2 \left(-\Delta - 2\lambda_i^{(k)} \right)^{-1} V^{(k)} \phi_i^{(k)}, \quad (8)$$

and orthogonalize $\widehat{\Phi} = (\hat{\phi}_1, \dots, \hat{\phi}_N)$ by calculating the Cholesky decomposition of the Gram matrix

$$\int_{\mathbb{R}^3} \widehat{\Phi}^T \widehat{\Phi} d\mathbf{r} = LL^T, \quad \widetilde{\Phi} = \widehat{\Phi} L^{-T}. \quad (9)$$

Then we calculate the $N \times N$ Fock matrix

$$F = \int_{\mathbb{R}^3} \widetilde{\Phi}^T H^{(k+1)} \widetilde{\Phi} d\mathbf{r}, \quad (10)$$

diagonalize F

$$\Lambda^{(k+1)} = S^{-1} F S, \quad (11)$$

and finally rotate the orbitals by S

$$\Phi^{(k+1)} = \widetilde{\Phi} S. \quad (12)$$

The new values of orbital energies are the diagonal parts of $\Lambda^{(k+1)}$. Steps of the iterative process are summarized in Algorithm 1

Algorithm 1 Block-Green iteration

- 1: Calculate $\widehat{\Phi} = (\hat{\phi}_1, \dots, \hat{\phi}_N)$: $\hat{\phi}_i = 2 \left(-\Delta - 2\lambda_i^{(k)} \right)^{-1} V^{(k)} \phi_i^{(k)}$.
 - 2: Orthogonalize $\widehat{\Phi}$: $\widetilde{\Phi} = \widehat{\Phi} L^{-T}$, where $L: LL^T = \int_{\mathbb{R}^3} \widehat{\Phi}^T \widehat{\Phi} d\mathbf{r}$.
 - 3: Calculate the Fock matrix $F = \int_{\mathbb{R}^3} \widetilde{\Phi}^T H^{(k+1)} \widetilde{\Phi} d\mathbf{r}$ via derivative-free formula in Statement 1.
 - 4: Find new orbital energies by diagonalizing F : $\Lambda^{(k+1)} = S^{-1} F S$
 - 5: Find new orbitals: $\Phi^{(k+1)} = \widetilde{\Phi} S$
-

4.2. Derivative-free Fock matrix computation

To compute the Fock matrix (10) one need to find Laplacian of orbital functions. Since all computations in tensor formats are done approximately, differentiation will lead to the accuracy loss. Therefore, we present the following derivative-free formula for the Fock matrix calculation

Statement 1. In Algorithm 1 the Fock matrix (10) can be written in the form without derivatives

$$F = \int_{\mathbb{R}^3} \left(\widetilde{\Phi}^T V^{(k+1)} \widetilde{\Phi} - \widetilde{\Phi}^T V^{(k)} \Phi^{(k)} L^{-T} + L^{-1} \widehat{\Phi}^T \widehat{\Phi} \Lambda^{(k)} L^{-T} \right) d\mathbf{r}, \quad (13)$$

where $V^{(k)} \equiv V(\Phi^{(k)})$.

Proof. From (10) we have

$$F = L^{-1} \left(\int_{\mathbb{R}^3} \widehat{\Phi}^T H^{(k+1)} \widehat{\Phi} d\mathbf{r} \right) L^{-T}, \quad (14)$$

Matrix

$$\int_{\mathbb{R}^3} \widehat{\Phi}^T H^{(k+1)} \widehat{\Phi} d\mathbf{r}$$

can be written elementwise in the following way

$$\begin{aligned} \int_{\mathbb{R}^3} \hat{\phi}_\alpha H^{(k+1)} \hat{\phi}_\beta dx &= \int_{\mathbb{R}^3} \hat{\phi}_\alpha \left(-\frac{1}{2} \Delta + V^{(k+1)} \right) \hat{\phi}_\beta dx = \\ &= \int_{\mathbb{R}^3} \hat{\phi}_\alpha V^{(k+1)} \hat{\phi}_\beta dx + \int_{\mathbb{R}^3} \hat{\phi}_\alpha \left(-\frac{1}{2} \Delta \pm \lambda_\beta^{(k)} \right) \hat{\phi}_\beta dx = \\ &= \int_{\mathbb{R}^3} \hat{\phi}_\alpha (V^{(k)} + \lambda_\beta^{(k)}) \hat{\phi}_\beta dx + \int_{\mathbb{R}^3} \hat{\phi}_\alpha \left(-\frac{1}{2} \Delta - \lambda_\beta^{(k)} \right) \hat{\phi}_\beta dx \end{aligned}$$

Taking into account that

$$\hat{\phi}_\beta = \left(-\frac{1}{2} \Delta - \lambda_\beta^{(k)} \right)^{-1} V^{(k)} \phi_\beta^{(k)}$$

we have

$$\int_{\mathbb{R}^3} \hat{\phi}_\alpha \left(-\frac{1}{2} \Delta - \lambda_\beta^{(k)} \right) \hat{\phi}_\beta dx = \int_{\mathbb{R}^3} \hat{\phi}_\alpha \left(-\frac{1}{2} \Delta - \lambda_\beta^{(k)} \right) \left(-\frac{1}{2} \Delta - \lambda_\beta^{(k)} \right)^{-1} V^{(k)} \phi_\beta^{(k)} dx = \int_{\mathbb{R}^3} \hat{\phi}_\alpha V^{(k)} \phi_\beta^{(k)} dx.$$

Finally, substituting the obtained expressions into (14) we get (13). ■

4.3. Density mixing

In order to accelerate convergence we used the density mixing scheme, which is typical used for DFT calculations [16]. Here we provide a brief description of the scheme for the sake of completeness. Our scheme is a fix point iteration which can be formally written as $\rho^{(k)} = G(\rho^{(k-1)})$. In the density mixing scheme, the next density $\rho^{(k+1)}$ is represented as a linear combination of the results of previous iterations

$$\rho^{(k+1)} = (1 - \beta_k) \sum_{j=1}^m \alpha_j \rho^{(k-m+j)} + \beta_k \sum_{j=1}^m \alpha_j G(\rho^{(k-m+j)}),$$

where the coefficients $\alpha = (\alpha_1, \dots, \alpha_m)$ are the solution of the minimization problem

$$\alpha = \arg \min_{\tilde{\alpha}_1, \dots, \tilde{\alpha}_m} \left\| \sum_{j=0}^m \tilde{\alpha}_j [\rho^{(k-m+j)} - G(\rho^{(k-m+j)})] \right\|$$

under the constraints

$$\sum_{j=0}^m \tilde{\alpha}_j = 1.$$

5. Discretization

It is known that the orbitals decay exponentially at infinity [21]:

$$\phi_i(\mathbf{r}) = O\left(e^{-\sqrt{-2\lambda_i} \|\mathbf{r}\|}\right), \quad \|\mathbf{r}\| \rightarrow +\infty. \quad (15)$$

As a result, multiplications and linear combinations of the orbitals also decay exponentially as $\|\mathbf{r}\| \rightarrow +\infty$. Hence, we replace the whole \mathbb{R}^3 with a finite box $\Omega = [-L, L]^3$, where L depends on the chosen accuracy and estimation of the smallest orbital energy λ_N . We will numerically investigate how the choice of L in the experiments section.

In Ω we introduce a uniform tensor-product grid $\omega^h = \omega_1^{h_1} \times \omega_2^{h_2} \times \omega_3^{h_3}$ with $h = 2L_i/n_i$, where $\omega_i^h = \{-L_i + kh_i : k = 0, \dots, n\}$, $i = 1, 2, 3$. We use uniform grids to obtain structured operators and significantly decrease computation complexity. Low-order approximation accuracy over uniform grids will be improved later by using extrapolation. For simplicity we use grids with $h_1 = h_2 = h_3$ and $L_1 = L_2 = L_3$. Algorithm (8)–(12) requires the computation of $V\Phi$ at each step, and the 3D-convolution have to be computed

$$w(\mathbf{r}) \equiv \int_{\mathbb{R}^3} \frac{f(\mathbf{r}')}{\|\mathbf{r} - \mathbf{r}'\|} d\mathbf{r}', \quad \mathbf{r} \in \mathbb{R}^3. \quad (16)$$

Remark 5.1. One can use the fact that problem of finding $w(\mathbf{r})$ is equivalent to find the solution of $-\Delta w = 4\pi f$ and it seems better to solve less expensive Poisson equation instead of direct computation of the convolution. Nevertheless, $w(\mathbf{r}) = O(1/\|\mathbf{r}\|)$ when $\|\mathbf{r}\| \rightarrow \infty$. Boundary conditions are unknown and, hence, to get the accuracy ϵ one have to choose domain size $L = O(1/\epsilon)$.

On the other hand we can compute convolution with the Yukawa kernel (operator $(-\Delta - 2\lambda)^{-1}$ in (8)) by directly inverting the shifted Laplacian. This can be done efficiently due to the exponential decay of orbitals (15), so it is safe to enforce Dirichlet boundary conditions. Moreover, the direct convolution with the Yukawa kernel discretized on a uniform grid leads to the non-symmetry of the Fock matrix and this breaks down the convergence. The point here is that the numerical analogue of the derivative-free formula (13) does not hold unless the action of $(-\Delta - 2\lambda)^{-1}$ is computed by solving screened Poisson equation.

Thus, to discretize $(-\Delta - 2\lambda)^{-1}$ we use standard 7-points finite difference scheme. To discretize convolutions (16) we use Galerkin scheme and piecewise-constant basis functions $\phi_{\mathbf{i}}$ with support on $\Omega_{\mathbf{i}}$, where $\mathbf{i} \in \mathcal{I} \equiv \{0, \dots, n-1\}^3$ and $\Omega_{\mathbf{i}}$ are h^3 cubes centered in $\mathbf{r}_{\mathbf{i}} = (x_i, y_j, z_k) \in \omega^h$. So,

$$w(\mathbf{r}_i) \equiv w_{\mathbf{i}} \approx \sum_{\mathbf{j} \in \mathcal{I}} f_{\mathbf{j}} q_{\mathbf{i}-\mathbf{j}}, \quad (17)$$

where

$$q_{i-j} = \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} \frac{\phi_i(\mathbf{r})\phi_j(\mathbf{r}')}{\|\mathbf{r} - \mathbf{r}'\|} d\mathbf{r}d\mathbf{r}', \quad f_i = \int_{\mathbb{R}^3} f(\mathbf{r})\phi_i(\mathbf{r}) d\mathbf{r}. \quad (18)$$

The total approximation error of the discretization scheme is $O(h^2)$. To get high-order discretization schemes translation-invariant basis functions of higher order can be used $\phi_i(\mathbf{r}) = \phi(\mathbf{r} - \mathbf{r}_i)$, where $\phi(\mathbf{r})$ is a suitable piecewise-polynomial function.

6. Operations in the Tucker format

In order to implement the iterative algorithm in the Tucker format, we need several basic operations. Linear operations (addition, multiplication by number) can be done straightforwardly [22]. Calculation of scalar products and norms can also be efficiently implemented within the tensor formats. After such operations the ranks typically increase, but can be reduced thanks to the SVD-based rounding procedure.

6.1. Elementwise operations and cross approximation

Despite linear complexity in the mode size, some operations such as elementwise products and convolutions may have strong rank dependence: $O(r^8)$ floating point operations. To decrease this complexity we use the so-called *cross approximation method* [23, 24, 25]. This method finds the decomposition using only few of elements of the approximated array. Every next iteration of the cross approximation procedure chooses adaptively new elements to be computed until the stopping criteria holds. In the implementation we use Schur-Cross3D algorithm proposed in [26] that has $O(r^3 + nr)$ complexity.

The cross approximation algorithm is also the key technique to approximate nonlinear elementwise functions of density arising in the calculation of exchange-correlation potential V_{xc} . We note, that $V_{xc}(\rho)$ itself does not have low-rank structure, however, $V_{xc}(\rho)\phi_i$ is of low rank.

6.2. Convolution

Computation of the convolution is one of the most expensive parts of (1) with strong dependence on ranks of input tensors. We calculate it by the *Cross-Conv* algorithm proposed in [26]. This algorithm has $O(r^4 + nr^2)$ complexity, which is the fastest known for practically interesting mode sizes n up to $n \sim 2^{15}$. However, for fine meshes it might be a good idea to use quantize tensor train (QTT) convolution algorithm [27] can be used. This algorithm computes the convolution of two tensors given in the Tucker format. The orbitals are already there, and it is well-known that the kernel function $1/r$ can be approximated in the Tucker format with $r = O(\log n \log^{-1} \epsilon)$. This trick with decomposition into sum of exponents was used to calculate external potential (3).

6.3. Fast Laplacian inversion

The Laplacian equation is solved using the algorithm proposed in [28]. Given the right-hand side in the Tucker format, the solution after the sine transform (which is separable) reduces to the computation of

$$\frac{a_{ijk}}{\eta_i + \eta_j + \eta_k - \mu},$$

where $\eta_i, i = 1, \dots, n$ are the eigenvalues of the one-dimensional Laplace operator. Such computation can be straightforwardly done using the cross-approximation technique.

7. Complexity estimates

For simplicity we provide an upper bound of complexity for r be the largest rank of all orbitals. We also assume that the calculations are done on $n \times n \times n$ grid. Precomputations before the iterative process include calculation of the external potential (3) and the Galerkin tensor of the convolution kernel (18). The computation of the external potential requires $N_{nuc} \cdot \mathcal{O}(r^3 + 3nr)$. Precomputation of the Galerkin tensor requires $\mathcal{O}(r^3 + 3nr)$ operations. It does not depend on number of nuclei but has a bigger constant due to one-dimensional integration.

Let us now estimate the complexity of one iteration. We denote by $K_{cross} = \mathcal{O}(r^4 + nr^2)$ the complexity of the cross approximation calculated by Schur-Cross3D and by $K_{conv} = \mathcal{O}(r^4 + nr^2 + rn \log n)$ the complexity of one convolution operation [26]. We note that the constant hidden in $\mathcal{O}(\cdot)$ is of the order of unity. Calculation of $V^{(k)}\phi_i^{(k)}$ in (8) is done as follows. First of all we calculate V_{coul} which is convolution operation that has K_{conv} complexity. The Coulomb potential is calculated only once on each iteration. In case of KS equations we run cross approximation procedure to calculate $V^{(k)}\phi_i^{(k)}$ since the V_{xc} does not have low-rank structure. For the HF equations we need additionally to calculate the exchange potential, which requires N^2 convolution operations and N^2 elementwise products. Thus, the step has NK_{cross} complexity for KS and N^2K_{cross} for the HF equations. The application of $(-\Delta - \mu)^{-1}$ is also of NK_{conv} complexity but with approximately twice smaller constant in K_{conv} as one of the input tensors has fixed rank 3.

Step (9) requires N^2 scalar product evaluations, so it has N^2K_{cross} complexity as elementwise products are done by the cross approximation. Complexity N^2K_{cross} is much larger than the Cholesky factorization as we assume that $N \ll r^4$. The Fock matrix computation (13) consists of $V^{(k)}\Phi$ which is of same complexity as $V^{(k)}\Phi^{(k)}$ and of scalar products. Thus, we get the overall complexity of one iteration to be $N^2 \cdot \mathcal{O}(r^4 + nr^2 + rn \log n)$.

8. Numerical experiments

The prototype is implemented in Python. The implementation of HF and KS solvers can be found at <https://github.com/rakhuba/tensorchem>. The toolbox with arithmetics in the Tucker format can be found at <https://github.com/rakhuba/tucker3d>. For the basic linear algebra tasks the MKL library is used. Python and MKL are from the Anaconda Python Distribution <https://store.continuum.io/cshop/anaconda/>. Python version is 2.7.9. MKL version is 11.1-1. Tests were performed on 4 Intel Core i7 2.6 GHz processor with 8GB of RAM. However, only 2 threads were used (this is default number of threads for MKL). We would like to emphasize that implementation of the whole algorithm is in Python and time performance can be improved by implementing the most time-consuming parts of it in C or Fortran languages.

8.1. The box size

Here we illustrate how the results depend on the simulation box size. The orbitals decay exponentially as $\exp(-\sqrt{-2\lambda_{HOMO}}\|x\|)$, $\|x\| \rightarrow \infty$. Therefore, one can expect that the error introduced by the finite size of the box has exponential decay with the box size. On Figure 8.1 illustrates this guess. On this figure the dependence of relative error of $E_h(a)$ with respect to the $E_h(\infty)$ as a function of a box size a is presented. All calculations were done for the fixed grid step $h = 0.1 \text{ \AA}$ and accuracy $\epsilon = 10^{-9}$. $E_h(\infty)$ was estimated at $a = 50 \text{ \AA}$.

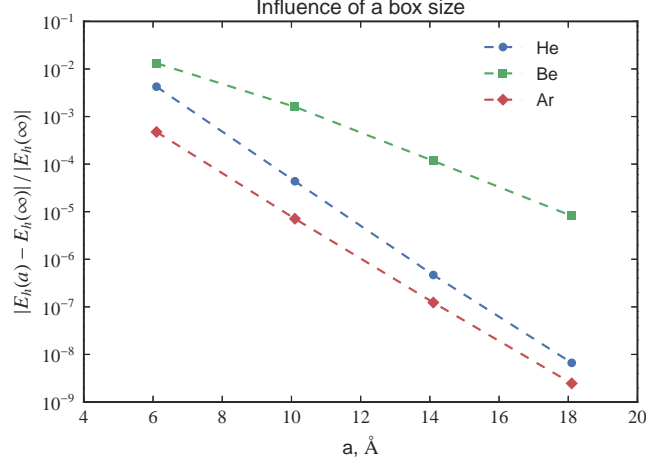


Figure 1: Dependence of relative error of $E_h(a)$ on a box size a for fixed $h = 0.1$ Å, $\epsilon = 10^{-9}$. To calculate $E_h(\infty)$ the box of size $a = 50$ Å was used.

8.2. Extrapolation

As the proposed approach is fully grid-based, a good idea is to start the method on a coarse grid with, say $N = 128$ and then use it as an initial guess on finer grids (grids with 2^k grid points are used). Due to the linear complexity of the proposed algorithm the total time for extrapolation is not larger than twice the time required for the computation on the finest grid:

$$t_{\text{extrapolated}} = t_N + t_{N/2} + \dots + t_{N/2^l} = CN + CN/2 + \dots + CN/2^l < 2CN = 2t_N, \quad (19)$$

where constant C does not depend on N . This makes the extrapolation a very useful and computationally efficient part of the whole algorithm.

Numerical results showed that on coarse grids the order of the approximation is not exactly of the second order. Hence, we used the so-called Aitken's delta squared process [29], which is equivalent to the Richardson extrapolation for the exact second order. The Aitken's process accelerates the convergence of the sequence E_n such that the new sequence E'_n

$$E'_n = E_{n+2} - \frac{(E_{n+2} - E_{n+1})^2}{E_{n+2} - 2E_{n+1} + E_n}, \quad (20)$$

converges faster than E_n when n goes to infinity. The same trick can be applied to E'_n . The results for extrapolation are shown on Figure 8.2.

8.3. Hartree-Fock calculations on atoms

First we present the HF calculations of closed-shell atoms He, Be, Ne, Ar. Table 8.3 represents total and highest occupied molecular orbital (HOMO) energies. These results are compared with highly accurate results for atoms [30]. Our grid-based results are obtained with the relative accuracy $\epsilon = 10^{-7}$. We use the relative error instead of absolute in order to obtain accurate HOMO values and at the same time to be faster in the calculation of lower orbitals. However,

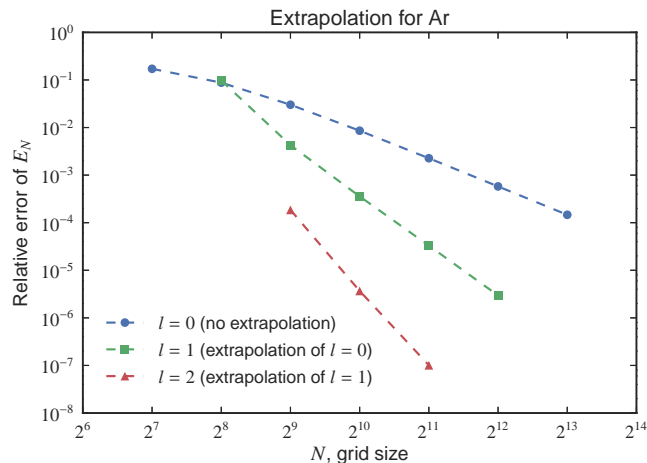


Figure 2: Extrapolation of the full energy as a function of grid size for Argon. Error is calculated with respect to highly accurate results from [30].

ϵ	10^{-3}	10^{-5}	10^{-7}	10^{-9}	10^{-11}
Total energy	-2.8615	-2.861678	-2.8616801	-2.861680000	-2.86167999593

Table 1: Helium total energy for different values of relative error ϵ . The energy was extrapolated on a sequence of grids: from 128^3 to 8196^3 . Bold denotes correct numbers. The total energy in [30] is -2.861 679 996.

absolute accuracy can also be used. Extrapolation is done on a range of grid sizes: from 128^3 to 8192^3 .

Results in Table 8.3 show systematic convergence of the total and HOMO energies. Note, that for all atoms under consideration extrapolation with grid sizes up to $N^3 = 8192^3$ grid points is enough to get accuracy $\epsilon = 10^{-7}$. Without extrapolation, much larger grids are needed.

8.4. LDA calculations on molecules

We perform the LDA calculations for several molecules. Molecular geometries were taken from the NIST database [32]. In Table 8.4 the resulting total and HOMO energies are compared with the LDA computations with the aug-cc-pVXZ (X=Q, 5) basis sets. The latter we done using NWCHEM program package [31]. In all experiments the relative error for all orbital functions was set to be $\epsilon = 10^{-7}$. The box size was chosen adaptively to the HOMO energy.

We note that the extrapolated energy is smaller than for the basis aug-cc-pV5Z approximately 10^{-4} hartree both for the total energy and for the HOMO energy, thus our method has less basis set error. Figure 8.4 illustrates rapid convergence of all five occupied orbitals with respect to the number of iterations for $\epsilon = 10^{-5}$. As anticipated the convergence stopped below ϵ . We note that in all experiments orbitals have converged to the $\epsilon = 10^{-7}$ within 30 iterations.

Atom	Method	Total energy	HOMO energy
He	$N = 8192$	-2.861 670	-0.917 950
	Extrapolated	-2.861 680	-0.917 955
	[30]	-2.861 680	-0.917 956
	aug-cc-pVQZ	-2.861 539	-0.917 915
	aug-cc-pV5Z	-2.861 635	-0.917 935
Be	$N = 8192$	-14.572 256	-0.309 263
	Extrapolated	-14.573 023	-0.309 269
	[30]	-14.573 023	-0.309 270
	aug-cc-pVQZ	-14.572 976	-0.309 269
	aug-cc-pV5Z	-14.573 011	-0.309 270
Ne	$N = 8192$	-128.518 74	-0.850 523
	Extrapolated	-128.547 08	-0.850 410
	[30]	-128.547 09	-0.850 410
	aug-cc-pVQZ	-128.544 69	-0.850 210
	aug-cc-pV5Z	-128.546 87	-0.850 391
Ar	$N = 8192$	-526.740 5	-0.591 024
	Extrapolated	-526.817 4	-0.590 017
	[30]	-526.817 5	-0.591 017
	aug-cc-pVQZ	-526.816 9	-0.591 013
	aug-cc-pV5Z	-526.817 3	-0.591 011

Table 2: Total and HOMO energies for different atoms, $\epsilon = 10^{-7}$. Extrapolated calculations were done on a sequence of grids: from 128^3 to 8196^3 . aug-cc-pVQZ and aug-cc-pV5Z calculations were done by NWCHEM [31].

Molecule	Method	Total energy	HOMO energy
H_2	$N = 8192$	-1.137 392 3	-0.378 667
	Extrapolated	-1.137 392 8	-0.378 668
	aug-cc-pVQZ	-1.137 249 9	-0.378 649
	aug-cc-pV5Z	-1.137 374 8	-0.378 665
CH_4	$N = 8192$	-40.116 452	-0.348 989
	Extrapolated	-40.119 829	-0.348 984
	aug-cc-pVQZ	-40.118 644	-0.348 964
	aug-cc-pV5Z	-40.119 299	-0.348 982
C_2H_6	$N = 8192$	-79.069 964	-0.299 763
	Extrapolated	-79.075 142	-0.299 761
	aug-cc-pVQZ	-79.070 784	-0.299 724
	aug-cc-pV5Z	-79.072 763	-0.299 762

Table 3: Total and HOMO energies for different molecules, relative error $\epsilon = 10^{-7}$. Extrapolated calculations were done on a sequence of grids: from 128^3 to 8196^3 . aug-cc-pVQZ and aug-cc-pV5Z calculations were done by NWCHEM [31].

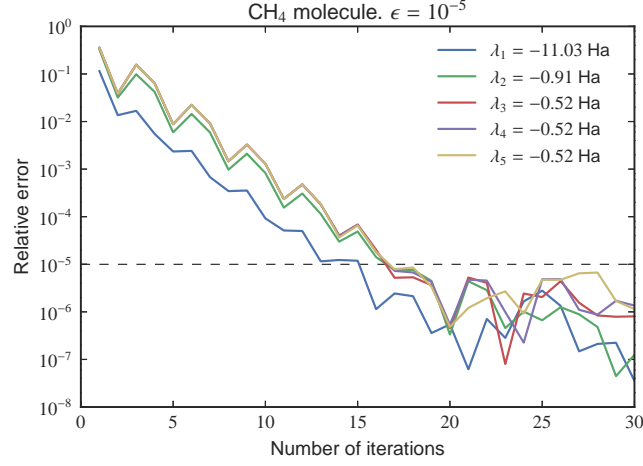


Figure 3: Relative errors of each orbital energy as a function of number of iterations ($\epsilon = 10^{-5}$).

8.5. LDA calculations on cluster of atoms

Here we present calculations on a test system of hydrogen atoms, which was firstly calculated in [13] as an artificial example. The system represents finite cluster of hydrogen atoms with unit cell having primitive cubic structure. The distance between atoms is set to be $d = 2 \text{ \AA}$. As in section with LDA calculations on molecules in all experiments we observed systematical convergence of the total energy with ϵ precision. Table 8.5 illustrates that ranks of orbitals grow sublinearly with the system size. Thus, we expect the proposed algorithm to be particularly efficient for systems with regular location of atoms. The time of one iteration on $N = 1024$ and $\epsilon = 10^{-5}$ is 9 sec for $H_{3 \times 2 \times 2}$ and 68 sec for $H_{8 \times 2 \times 2}$, which as anticipated scales approximately quadratically with the size of the system: $(8/3)^2 \approx 7.1$ and $68/9 \approx 7.5$.

9. Conclusions and future work

Tensor approach to the solution of 3D problems in electronic computations allows us to reach desirable accuracy at moderate computational cost. This can be used, for example, in verification of other methods and in construction of accurate basis sets for other methods [33]. The main problem with the presented approach is still in its scaling for large N : for example, one has to compute the approximation of all the products $\phi_i \phi_j$ using the cross method. This can be solved by the resolution of identity methods [34], but they are typically used for global basis sets. What is interesting, is that the density itself shows a very good low-rank structure; thus, it is very interesting to apply tensor decomposition methods for the orbital-free DFT functionals [35, 36].

10. Acknowledgements

We thank DrSci. Boris Khoromskij and Dr. Venera Khoromskaia for useful comments on the draft of the manuscript.

Cluster	min ranks orbital	max ranks orbital
$H_{1 \times 2 \times 2}$	$16 \times 16 \times 14$	$17 \times 17 \times 15$
$H_{3 \times 2 \times 2}$	$28 \times 28 \times 21$	$35 \times 35 \times 20$
$H_{8 \times 2 \times 2}$	$25 \times 25 \times 22$	$36 \times 36 \times 26$
$H_{1 \times 2 \times 1}$	$13 \times 12 \times 12$	$13 \times 12 \times 12$
$H_{2 \times 2 \times 1}$	$16 \times 16 \times 14$	$17 \times 17 \times 15$
$H_{5 \times 2 \times 1}$	$16 \times 18 \times 11$	$16 \times 20 \times 13$
$H_{9 \times 2 \times 1}$	$14 \times 18 \times 11$	$17 \times 22 \times 13$
$H_{16 \times 2 \times 1}$	$16 \times 19 \times 11$	$21 \times 27 \times 14$
$H_{1 \times 4 \times 1}$	$13 \times 12 \times 13$	$13 \times 13 \times 13$
$H_{3 \times 4 \times 1}$	$19 \times 20 \times 13$	$23 \times 23 \times 15$
$H_{8 \times 4 \times 1}$	$23 \times 20 \times 12$	$32 \times 27 \times 15$

Table 4: Ranks of orbitals with maximum and minimum ranks on different clusters of hydrogen atoms.

References

- [1] D. R. Hartree, The calculation of atomic structures, J. Wiley, 1957.
- [2] W. Kohn, L. J. Sham, Self-consistent equations including exchange and correlation effects, Phys. Rev. 140 (4A) (1965) A1133.
- [3] R. J. Harrison, G. I. Fann, T. Yanai, Z. Gan, G. Beylkin, Multiresolution quantum chemistry: Basic theory and initial applications, J. Chem. Phys. 121 (23) (2004) 11587–11598.
- [4] P. Motamarri, M. R. Nowak, K. Leiter, J. Knap, V. Gavini, Higher-order adaptive finite-element methods for Kohn–Sham density functional theory, J. Comput. Phys. 253 (2013) 308–343.
- [5] J. R. Chelikowsky, N. Troullier, Y. Saad, Finite-difference-pseudopotential method: electronic structure calculations without a basis, Phys. Rev. Lett. 72 (8) (1994) 1240.
- [6] J. J. Mortensen, L. B. Hansen, K. W. Jacobsen, Real-space grid implementation of the projector augmented wave method, Phys. Rev. B 71 (3) (2005) 035109.
- [7] V. Khoromskaia, B. N. Khoromskij, R. Schneider, Tensor-structured factorized calculation of two-electron integrals in a general basis, SIAM J. Sci. Comput. 35 (2) (2013) A987–A1010. doi:10.1137/120884067.
- [8] V. Khoromskaia, Computation of the Hartree-Fock exchange by tensor-structured methods, Comput. Methods Appl. Math. 10 (2).
- [9] B. N. Khoromskij, V. Khoromskaia, S. R. Chinnamsetty, H.-J. Flad, Tensor decomposition in electronic structure calculations on 3D Cartesian grids, J. Comput. Phys. 228 (16) (2009) 5749–5762. doi:10.1016/j.jcp.2009.04.043.
- [10] V. Khoromskaia, Numerical solution of the Hartree-Fock equation by multilevel tensor-structured methods, Ph.D. thesis, TU Berlin (2010). URL <http://opus.kobv.de/tuberlin/volltexte/2011/2948/>
- [11] B. N. Khoromskij, V. Khoromskaia, H.-J. Flad., Numerical solution of the Hartree-Fock equation in multilevel tensor-structured format, SIAM J. Sci. Comput. 33 (1) (2011) 45–65. doi:10.1137/090777372.
- [12] V. Khoromskaia, B. N. Khoromskij, R. Schneider, QTT representation of the Hartree and exchange operators in electronic structure calculations, Comput. Methods Appl. Math 11 (3) (2011) 327–341.
- [13] V. Khoromskaia, Black-box Hartree-Fock solver by tensor numerical methods, Computational Methods in Applied Mathematics 14 (1) (2014) 89–111. doi:10.1515/cmam-2013-0023.
- [14] B. N. Khoromskij, V. Khoromskaia, Multigrid accelerated tensor approximation of function related multidimensional arrays, SIAM J. Sci. Comput. 31 (4) (2009) 3002–3026. doi:10.1137/080730408.
- [15] V. Khoromskaia, B. N. Khoromskij, Tensor numerical methods in quantum chemistry: from hartree-fock to excitation energies, Physical Chemistry Chemical Physics.
- [16] F. Jensen, Introduction to computational chemistry, John Wiley & Sons, 2013.
- [17] J. P. Perdew, A. Zunger, Self-interaction correction to density-functional approximations for many-electron systems, Phys. Rev. B 23 (10) (1981) 5048.
- [18] L. Tucker, Implications of factor analysis of three-way matrices for measurement of change, Problems in measuring change (1963) 122–137.

- [19] L. de Lathauwer, B. de Moor, J. Vandewalle, A multilinear singular value decomposition, *SIAM J. Matrix Anal. Appl.* 21 (2000) 1253–1278. doi:10.1137/s0895479896305696.
- [20] B. N. Khoromskij, V. Khoromskaia, Low rank Tucker-type tensor approximation to classical potentials, *Central European journal of mathematics* 5 (3) (2007) 523–550. doi:10.2478/s11533-007-0018-0.
- [21] T. Ishida, K. Ohno, On the asymptotic behavior of Hartree-Fock orbitals, *Theoretica Chimica Acta* 81 (6) (1992) 355–364.
- [22] I. V. Oseledets, D. V. Savostyanov, E. E. Tyrtyshnikov, Linear algebra for tensor problems, *Computing* 85 (3) (2009) 169–188. doi:10.1007/s00607-009-0047-6.
- [23] E. E. Tyrtyshnikov, Incomplete cross approximation in the mosaic-skeleton method, *Computing* 64 (4) (2000) 367–380. doi:10.1007/s006070070031.
- [24] M. Bebendorf, Approximation of boundary element matrices, *Numer. Mathem.* 86 (4) (2000) 565–589. doi:10.1007/p100005410.
- [25] I. V. Oseledets, D. V. Savostyanov, E. E. Tyrtyshnikov, Tucker dimensionality reduction of three-dimensional arrays in linear time, *SIAM J. Matrix Anal. Appl.* 30 (3) (2008) 939–956. doi:10.1137/060655894.
- [26] M. V. Rakhuba, I. V. Oseledets, Fast multidimensional convolution in low-rank tensor formats via cross approximation, *SIAM J. Sci. Comput.* 37 (2) (2015) A565–A582. doi:10.1137/140958529.
- [27] V. Kazeev, B. Khoromskij, E. Tyrtyshnikov, Multilevel Toeplitz matrices generated by tensor-structured vectors and convolution with logarithmic complexity, *SIAM J. Sci. Comput.* 35 (3) (2013) A1511–A1536. doi:10.1137/110844830.
- [28] I. V. Oseledets, E. A. Muravleva, Fast orthogonalization to the kernel of discrete gradient operator with application to the Stokes problem, *Linear Algebra Appl.* 432 (6) (2010) 1492–1500. doi:10.1016/j.laa.2009.11.010.
- [29] D. C. Joyce, Survey of extrapolation processes in numerical analysis, *SIAM Rev.* 13 (4) (1971) 435–490.
- [30] T. Koga, S. Watanabe, K. Kanayama, R. Yasuda, A. J. Thakkar, Improved Roothaan–Hartree–Fock wave functions for atoms and ions with $n \leq 54$, *J. Chem. Phys.* 103 (8) (1995) 3000–3005.
- [31] M. Valiev, E. J. Bylaska, N. Govind, K. Kowalski, T. P. Straatsma, H. J. Van Dam, D. Wang, J. Nieplocha, E. Apra, T. L. Windus, et al., Nwchem: a comprehensive and scalable open-source solution for large scale molecular simulations, *Comput. Phys. Commun.* 181 (9) (2010) 1477–1489.
- [32] J. I. Russell D., *NIST Computational Chemistry Comparison and Benchmark Database Number 101 Release 16a*. URL <http://webbook.nist.gov>
- [33] Ö. Legeza, T. Rohwedder, R. Schneider, S. Szalay, Tensor product approximation (dmrg) and coupled cluster method in quantum chemistry, in: *Many-Electron Approaches in Physics, Chemistry and Mathematics*, Springer, 2014, pp. 53–76.
- [34] O. Vahtras, J. Almlöf, M. Feyereisen, Integral approximations for lcao-scf calculations, *Chem. Phys. Lett.* 213 (5) (1993) 514–518.
- [35] V. Gavini, K. Bhattacharya, M. Ortiz, Quasi-continuum orbital-free density-functional theory: A route to multi-million atom non-periodic DFT calculation, *J. Mech. Phys. Solids* 55 (4) (2007) 697–718.
- [36] T. Blesgen, V. Gavini, V. Khoromskaia, Approximation of the electron density of aluminium clusters in tensor-product format, *J. Comput. Phys.* 231 (6) (2012) 2551–2564.